



Real-Time Optimization for Economic Model Predictive Control

Sokoler, Leo Emil; Edlund, Kristian; Frison, Gianluca; Skajaa, Anders; Jørgensen, John Bagterp

Published in:

The 10th European Workshop on Advanced Control and Diagnosis (ACD 2012)

Publication date:

2012

[Link back to DTU Orbit](#)

Citation (APA):

Sokoler, L. E., Edlund, K., Frison, G., Skajaa, A., & Jørgensen, J. B. (2012). Real-Time Optimization for Economic Model Predictive Control. In *The 10th European Workshop on Advanced Control and Diagnosis (ACD 2012)* Technical University of Denmark.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Real-Time Optimization for Economic Model Predictive Control

Leo Emil Sokoler^{*,**} Kristian Edlund^{**} Gianluca Frison^{*}
Anders Skajaa^{*} John Bagterp Jørgensen^{*}

^{*} DTU Informatics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark (*{leso,giaf,andsk,jbj}@imm.dtu.dk*)

^{**} DONG Energy, DK-2820 Gentofte, Denmark (*kried@dongenergy.dk*)

Abstract: In this paper, we develop an efficient homogeneous and self-dual interior-point method for the linear programs arising in economic model predictive control. To exploit structure in the optimization problems, the algorithm employs a highly specialized Riccati iteration procedure. Simulations show that in comparison to conventional interior-point methods, our solver is a) significantly faster per. iteration and b) converges in a smaller and less fluctuating number of iterations.

1. INTRODUCTION

The main computational task in economic model predictive control (MPC) is real-time minimization of an economic objective function subject to system dynamics, input limits, input-rate limits and soft output limits. To make large-scale linear programs of this type solvable in real-time, we present a variant of Mehrotra's predictor-corrector interior-point method (IPM) (Mehrotra, 1992; Wright, 1987; Czyzyk et al., 1999) that combines a homogeneous and self-dual model (Andersen et al., 1996; Xu et al., 1996; Ye et al., 1994; Jansen et al., 1993) with a Riccati iteration procedure (Rao et al., 1998; Jørgensen et al., 2004; Wang and Boyd, 2010; Jørgensen et al., 2012). The algorithm is tested in a small conceptual example where economic MPC is applied to minimize operational costs of a power system. In addition, we compare our algorithm against state of the art general purpose solvers and a conventional structure-exploiting IPM.

1.1 Related Work

Structure-exploiting IPMs for set-point based MPC with ℓ_2 -penalty have been reported in (Rao et al., 1998; Jørgensen et al., 2004; Wang and Boyd, 2010; Jørgensen et al., 2012), and similar work for ℓ_1 -penalty in (Edlund et al., 2009; Vandenberghe et al., 2002). For general small and medium dimensional convex optimal control problems, custom code generators such as CVXGEN (Mattingley et al., 2010) facilitate MPC for systems with dynamics even in the kHz range. First order methods aimed at MPC for embedded systems have been described by (Bemporad and Patrino, 2012; Hans-Bernd and Ebenbauer, 2012; Jones et al., 2012). Another branch of emerging algorithms for fast MPC utilizes parametric programming (Alessio and Bemporad, 2009; Kvasnica and Fikar, 2010). Traditionally, such algorithms have been limited to small systems due to an exponential growth in complexity. However, work inspired by this field such as qpOASES (Potschka et al., 2010; Ferreau et al., 2008, 2012), partial enumeration (Pannocchia et al., 2007) and the multiresolution approximation method (Summers et al., 2011) can be applied

in real-time to larger systems. For parallelization of the online algorithm, iterative methods such as (Hartley et al., 2012; Kerrigan et al., 2012) as well as the alternating direction multiplier method (Markus and Findeisen, 2012) have been proposed.

2. PAPER ORGANIZATION

This paper is organized as follows. In Section 3, we formulate the linear program solved in economic MPC and optimality conditions are derived. Section 4 presents a Riccati iteration procedure for specializing a homogeneous and self-dual IPM to economic MPC. A case study of economic MPC and benchmarks are provided in Section 5. We give concluding remarks in section 6.

3. PROBLEM DEFINITION

The optimization problem solved in economic MPC may be stated as

$$\min_{u,x,y,w} \phi(u,x,w) = \sum_{k=0}^{N-1} p_k^T u_k + q_{k+1}^T w_{k+1}, \quad (1a)$$

$$\text{s.t. } x_{k+1} = Ax_k + Bu_k, \quad (1b)$$

$$y_{k+1} = Cx_{k+1}, \quad (1c)$$

$$\underline{u}_k \leq u_k \leq \bar{u}_k, \quad (1d)$$

$$\Delta \underline{u}_k \leq u_k - Dx_k \leq \Delta \bar{u}_k, \quad (1e)$$

$$\underline{y}_{k+1} - w_{k+1} \leq y_{k+1} \leq \bar{y}_{k+1} + w_{k+1}, \quad (1f)$$

$$w_{k+1} \geq 0, \quad (1g)$$

for $k \in \mathcal{N} := \{0, 1, \dots, N-1\}$. The problem data are the state space matrices (A, B, C) , the initial state x_0 , the input limits \underline{u}_k and \bar{u}_k , the input-rate limits $\Delta \underline{u}_k$ and $\Delta \bar{u}_k$, the output limits \underline{y}_k and \bar{y}_k , the input prices p_k , the prices for violating the output limits q_k and the prediction horizon N . We have labelled the inputs as $u_k \in \mathbb{R}^{n_u}$, the states as $x_k \in \mathbb{R}^{n_x}$, the outputs as $y_k \in \mathbb{R}^{n_y}$ and the soft variables as $w_k \in \mathbb{R}^{n_w}$. It is assumed that the state space system has been augmented such that $Dx_k = u_{k-1}$.

3.1 Linear Program Formulation

In a simple form, we can write (1) as

$$\min_t g^T t, \quad (2a)$$

$$\text{s.t. } Ft = b, \quad (2b)$$

$$Ht \leq c. \quad (2c)$$

E.g. for $N = 2$ the structures are

$$t := [u_0^T \ x_1^T \ w_1^T \ u_1^T \ x_2^T \ w_2^T]^T, \\ g := [p_0^T \ 0 \ q_1^T \ p_1^T \ 0 \ q_2^T]^T,$$

and

$$[F|b] := \left[\begin{array}{cccccc|c} B & -I & 0 & 0 & 0 & 0 & -Ax_0 \\ 0 & A & 0 & B & -I & 0 & 0 \end{array} \right], \\ [H|c] := \left[\begin{array}{cccccc|c} I & 0 & 0 & 0 & 0 & 0 & \bar{u}_0 \\ 0 & 0 & 0 & I & 0 & 0 & \bar{u}_1 \\ -I & 0 & 0 & 0 & 0 & 0 & -\underline{u}_0 \\ 0 & 0 & 0 & -I & 0 & 0 & -\underline{u}_1 \\ I & 0 & 0 & 0 & 0 & 0 & \Delta \bar{u}_0 \\ 0 & -D & 0 & I & 0 & 0 & \Delta \bar{u}_1 \\ -I & 0 & 0 & 0 & 0 & 0 & -\Delta \underline{u}_0 \\ 0 & D & 0 & -I & 0 & 0 & -\Delta \underline{u}_1 \\ 0 & C & -I & 0 & 0 & 0 & \bar{y}_1 \\ 0 & 0 & 0 & 0 & C & -I & \bar{y}_2 \\ 0 & -C & -I & 0 & 0 & 0 & -\underline{y}_1 \\ 0 & 0 & 0 & 0 & -C & -I & -\underline{y}_2 \\ 0 & 0 & -I & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -I & 0 \end{array} \right],$$

where we have defined

$$\Delta \bar{u}_0 := \Delta \bar{u}_0 + Dx_0, \\ \Delta \underline{u}_0 := \Delta \underline{u}_0 + Dx_0.$$

The optimization problem is a highly structured linear program with $n := N(n_u + n_x + n_y)$ variables, $m_E := Nn_x$ equality constraints and $m_I := N(4n_u + 3n_y)$ inequality constraints.

3.2 Optimality Conditions

The Lagrangian for (2) is given by

$$\mathcal{L}(t, p, z) := g^T t - p^T (b - Ft) - z^T (c - Ht).$$

Hence, the first order necessary and sufficient optimality conditions may be stated as

$$g + F^T p + H^T z = 0, \quad (3a)$$

$$b - Ft = 0, \quad (3b)$$

$$c - Ht - s = 0, \quad (3c)$$

$$z_i s_i = 0, \quad i = 1, 2, \dots, m_I, \quad (3d)$$

$$(z_i, s_i) \geq 0, \quad i = 1, 2, \dots, m_I, \quad (3e)$$

in which we have introduced the dual variables $p \in \mathbb{R}^{m_E}$ and $z \in \mathbb{R}^{m_I}$, and the slack variables $s := c - Ht$. Now define Z as the diagonal matrix with diagonal elements z_1, \dots, z_{m_I} and similarly for S . Let $\mathbf{1}_{m_I}$ be the vector of all ones of size m_I . We can then state the optimality conditions as

$$V(t, p, z, s) := \begin{bmatrix} g + F^T p + H^T z \\ b - Ft \\ c - Ht - s \\ ZS\mathbf{1}_{m_I} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (4)$$

subject to $(z, s) \geq 0$.

4. INTERIOR POINT METHODS

We now present a variant of Mehrotra's predictor-corrector IPM (Mehrotra, 1992; Wright, 1987; Czyzyk et al., 1999) for solving (2). The algorithm loosely tracks the central path by combining an affine step and a predictor-corrector step. The central path is defined as the set of points

$$\mathcal{C} := \left\{ \begin{bmatrix} t \\ p \\ z \\ s \end{bmatrix} \mid V(t, p, z, s) = \begin{bmatrix} \sigma(g + F^T p^0 + H^T z^0) \\ \sigma(b - Ft^0) \\ \sigma(c - Ht^0 - s^0) \\ \sigma \mu^0 e \end{bmatrix} \right\},$$

where $\sigma \in [0, 1]$ and $\mu^0 = (z^0)^T s^0 / m_I$. This path connects a given initial point (t^0, p^0, z^0, s^0) where $(z^0, s^0) \geq 0$, to a solution (t^*, p^*, z^*, s^*) satisfying (4). At iteration k , the affine direction $(\sigma = 0)$ is determined by solving

$$J_V(t^k, p^k, z^k, s^k) \begin{bmatrix} \Delta \tilde{t}^k \\ \Delta \tilde{p}^k \\ \Delta \tilde{z}^k \\ \Delta \tilde{s}^k \end{bmatrix} = -V(t^k, p^k, z^k, s^k). \quad (5)$$

The Jacobian of V evaluated at (t^k, p^k, z^k, s^k) is given by

$$J_V(t^k, p^k, z^k, s^k) = \begin{bmatrix} 0 & F^T & H^T & 0 \\ -F & 0 & 0 & 0 \\ -H & 0 & 0 & -I \\ 0 & 0 & S^k & Z^k \end{bmatrix}.$$

For the predictor-corrector step ($\sigma > 0$), we solve (5) with the modified right hand side

$$\tilde{V}(t^k, p^k, z^k, s^k) := \begin{bmatrix} (1 - \sigma^k)(g + F^T p^k + H^T z^k) \\ (1 - \sigma^k)(b - Ft^k) \\ (1 - \sigma^k)(c - Ht^k - s^k) \\ (Z^k S^k + \Delta \tilde{Z}^k \Delta \tilde{S}^k) \mathbf{1}_{m_I} - \sigma^k \mu^k \mathbf{1}_{m_I} \end{bmatrix}.$$

Here $\Delta \tilde{Z}^k$ and $\Delta \tilde{S}^k$ are second order correction terms defined as diagonal matrices with diagonal elements $\tilde{z}_1^k, \dots, \tilde{z}_{m_I}^k$ and $\tilde{s}_1^k, \dots, \tilde{s}_{m_I}^k$. To update σ^k , we use a heuristic described in (Nocedal and Wright, 1999) that compares the affine duality gap $\tilde{\mu}^k$ with the current duality gap μ^k

$$\sigma^k := (\tilde{\mu}^k / \mu^k)^3 = ((\tilde{z}^k)^T \tilde{s}^k / m_I) / ((z^k)^T s^k / m_I)^3,$$

where $\tilde{z}^k := z^k + \tilde{\alpha}^k \Delta \tilde{z}^k$ and $\tilde{s}^k := s^k + \tilde{\beta}^k \Delta \tilde{s}^k$. The parameters $\tilde{\alpha}$ and $\tilde{\beta}$ are used to ensure that $(\tilde{z}^k, \tilde{s}^k) \geq 0$

$$\tilde{\alpha}^k := \max\{\tilde{a} \in [0, 1] \mid \tilde{z}^k + \tilde{a} \Delta \tilde{z}^k \geq 0\},$$

$$\tilde{\beta}^k := \max\{\tilde{b} \in [0, 1] \mid \tilde{s}^k + \tilde{b} \Delta \tilde{s}^k \geq 0\}.$$

By updating σ as described above, the search direction is forced towards the central path if $\tilde{\mu}^k \approx \mu^k$, meaning that only small progress towards the solution can be made in the affine direction. We have summarized the IPM outlined above in Algorithm 1. To keep the iterates away from the boundary of the feasible region, this implementation includes a damping parameter ν in the range $[0.95; 0.999]$. For stopping criterions, we use

$$\frac{\|b - Ft^k\|_\infty}{1 + \|b\|_\infty} \leq \epsilon_E, \\ \frac{\|c - Ht^k - s^k\|_\infty}{1 + \|c\|_\infty} \leq \epsilon_I, \\ \frac{\|g + F^T p^k + H^T z^k\|_\infty}{1 + \|g\|_\infty} \leq \epsilon_D, \\ \frac{|g^T t^k - (-b^T p^k - c^T z^k)|}{1 + |-b^T p^k - c^T z^k|} \leq \epsilon_G.$$

Here ϵ_E , ϵ_I , ϵ_D and ϵ_G are small user-defined tolerances representing accuracy in primal and dual feasibility, as well as accuracy in the duality gap.

Algorithm 1 Interior-point algorithm for (2)

Require: $\begin{cases} \text{DATA} & (g, F, b, H, c) \\ \text{INITIAL POINT} & (t, p, z, s) \\ \text{PARAMETERS} & \nu \end{cases}$

```

// initialize
 $\mu \leftarrow z^T s / m_I$ 
while CONVERGED do
  // affine step
   $(\Delta \tilde{t}, \Delta \tilde{p}, \Delta \tilde{z}, \Delta \tilde{s}) \leftarrow -J_V(t, p, z, s)^{-1} V(t^k, p^k, z^k, s^k)$ 
  // center parameter
   $\tilde{\alpha} \leftarrow \max\{\tilde{a} \in [0, 1] | z + \tilde{a} \Delta \tilde{z} \geq 0\}$ 
   $\tilde{\beta} \leftarrow \max\{\tilde{b} \in [0, 1] | s + \tilde{b} \Delta \tilde{s} \geq 0\}$ 
   $\tilde{s} \leftarrow s + \tilde{\alpha} \Delta \tilde{s}$ 
   $\tilde{z} \leftarrow z + \tilde{\beta} \Delta \tilde{z}$ 
   $\tilde{\mu} \leftarrow \tilde{z}^T \tilde{s} / m_I$ 
   $\sigma \leftarrow (\tilde{\mu} / \mu)^3$ 
  // predictor-corrector step
   $(\Delta t, \Delta p, \Delta z, \Delta s) \leftarrow -J_V(t, p, z, s)^{-1} \tilde{V}(t^k, p^k, z^k, s^k)$ 
  // step update
   $\alpha \leftarrow \max\{a \in [0, 1] | z + a \Delta z \geq 0\}$ 
   $\beta \leftarrow \max\{b \in [0, 1] | s + b \Delta s \geq 0\}$ 
   $x \leftarrow t + \nu \alpha \Delta t$ 
   $s \leftarrow s + \nu \alpha \Delta s$ 
   $p \leftarrow p + \nu \beta \Delta p$ 
   $z \leftarrow z + \nu \beta \Delta z$ 
   $\mu \leftarrow z^T s / m_I$ 
end while return  $(t, p, z, s)$ 

```

4.1 Riccati Iteration Procedure

The main operations in Algorithm 1 consist of solving linear systems in the form

$$\begin{bmatrix} 0 & F^T & H^T & 0 \\ -F & 0 & 0 & 0 \\ -H & 0 & 0 & -I \\ 0 & 0 & S & Z \end{bmatrix} \begin{bmatrix} \Delta t \\ \Delta p \\ \Delta z \\ \Delta s \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ r_I \\ r_C \end{bmatrix}. \quad (6)$$

By eliminating Δs using the last row, we obtain the reduced system

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & Z^{-1}S \end{bmatrix} \begin{bmatrix} \Delta t \\ \Delta p \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ \hat{r}_I \end{bmatrix}, \quad (7)$$

in which $\hat{r}_I := r_I + Z^{-1}r_C$. We now show that this system can be solved efficiently for the optimization problem (1). The Lagrange multipliers associated with the inequality constraints (1d-1g) are labelled $\Delta\eta$, $\Delta\lambda$, Δv , $\Delta\omega$, $\Delta\gamma$, $\Delta\rho$ and $\Delta\xi$ where

$$\Delta\eta := [\Delta\eta_0^T \ \Delta\eta_1^T \ \dots \ \Delta\eta_{N-1}^T]^T$$

and similarly for $\Delta\lambda$, Δv , $\Delta\omega$, $\Delta\gamma$, $\Delta\rho$ and $\Delta\xi$. Here $\Delta\eta$ and $\Delta\lambda$ are multipliers for the input limits (1d), Δv and $\Delta\omega$ are multipliers for the input-rate limits (1e), $\Delta\gamma$ and $\Delta\rho$ are multipliers for the output limits (1f) and $\Delta\xi$ is the vector of multipliers for the non-negative constraints (1g). Based on this notation, the optimization variables in the system (7) can be expressed as

$$\Delta t = [\Delta u_0^T \ \Delta x_1^T \ \Delta w_1^T \ \dots \ \Delta u_{N-1}^T \ \Delta x_N^T \ \Delta w_N^T]^T,$$

$$\Delta p = [\Delta p_0^T \ \Delta p_1^T \ \dots \ \Delta p_{N-1}^T]^T,$$

$$\Delta z = [\Delta \eta^T \ \Delta \lambda^T \ \Delta v^T \ \Delta \omega^T \ \Delta \gamma^T \ \Delta \rho^T \ \Delta \xi^T]^T.$$

Accordingly, we partition the right hand side such that

$$r_D = [r_{u,0}^T \ r_{x,1}^T \ r_{w,1}^T \ \dots \ r_{u,N-1}^T \ r_{x,N}^T \ r_{w,N}^T]^T,$$

$$r_E = [r_{p,0}^T \ r_{p,1}^T \ \dots \ r_{p,N-1}^T]^T,$$

$$\hat{r}_I = [r_\eta^T \ r_\lambda^T \ r_v^T \ r_\omega^T \ r_\gamma^T \ r_\rho^T \ r_\xi^T]^T,$$

and write the diagonal matrix $Z^{-1}S$ in terms of diagonal submatrices

$$Z^{-1}S = \text{diag}(\Sigma_\eta^T, \Sigma_\lambda^T, \Sigma_v^T, \Sigma_\omega^T, \Sigma_\gamma^T, \Sigma_\rho^T, \Sigma_\xi^T).$$

The linear system of equations (7) may now be stated in the form

$$\begin{aligned} \Delta\eta_i - \Delta\lambda_i + \Delta v_i - \Delta\omega_i + B^T \Delta p_i &= r_{u,i}, & i \in \mathcal{N}, \\ -\Delta u_i + \Sigma_{\eta,i} \Delta\eta_i &= r_{\eta,i}, & i \in \mathcal{N}, \\ \Delta u_i + \Sigma_{\lambda,i} \Delta\lambda_i &= r_{\lambda,i}, & i \in \mathcal{N}, \\ -\Delta u_i + D \Delta x_i + \Sigma_{v,i} \Delta v_i &= r_{v,i}, & i \in \bar{\mathcal{N}}, \\ \Delta u_i - D \Delta x_i + \Sigma_{\omega,i} \Delta\omega_i &= r_{\omega,i}, & i \in \bar{\mathcal{N}}, \\ \Delta x_{i+1} - A \Delta x_i - B \Delta u_i &= r_{p,i}, & i \in \bar{\mathcal{N}}, \\ \Delta w_{i+1} - C \Delta x_{i+1} + \Sigma_{\gamma,i+1} \Delta\gamma_{i+1} &= r_{\gamma,i+1}, & i \in \mathcal{N}, \\ \Delta w_{i+1} + C \Delta x_{i+1} + \Sigma_{\rho,i+1} \Delta\rho_{i+1} &= r_{\rho,i+1}, & i \in \mathcal{N}, \\ \Delta w_{i+1} + \Sigma_{\xi,i+1} \Delta\xi_{i+1} &= r_{\xi,i+1}, & i \in \mathcal{N}, \\ -\Delta\gamma_{i+1} - \Delta\rho_{i+1} - \Delta\xi_{i+1} &= r_{w,i+1}, & i \in \mathcal{N}, \\ -\Delta p_i + C^T(\Delta\gamma_{i+1} - \Delta\rho_{i+1}) + \\ A^T \Delta p_i + D^T(\omega_i - \Delta v_i) &= r_{x,i}, & i \in \bar{\mathcal{N}}, \end{aligned}$$

with $\bar{\mathcal{N}} := \mathcal{N} \setminus 0$, and the special cases

$$-\Delta u_0 + \Sigma_{v,0} \Delta v_0 = r_{v,0},$$

$$\Delta u_0 + \Sigma_{\omega,0} \Delta\omega_0 = r_{\omega,0},$$

$$\Delta x_1 - B \Delta u_0 = r_{p,0},$$

$$-\Delta p_{N-1} + C^T(\Delta\gamma_N - \Delta\rho_N) = r_{x,N}.$$

By eliminating the Lagrange multipliers for the inequality constraints $\Delta\eta$, $\Delta\lambda$, Δv , $\Delta\omega$, $\Delta\gamma$, $\Delta\rho$ and $\Delta\xi$ we get the reduced set of equations

$$B^T \Delta p_0 + U_0 \Delta u_0 = R_{u,0}, \quad (8a)$$

$$B^T \Delta p_i + U_i \Delta u_i + G_i \Delta x_i = R_{u,i}, i \in \bar{\mathcal{N}} \quad (8b)$$

$$-\Delta x_1 + B \Delta u_0 = R_{p,0}, \quad (8c)$$

$$-\Delta x_{i+1} + A \Delta x_i + B \Delta u_i = R_{p,i}, i \in \bar{\mathcal{N}}, \quad (8d)$$

$$W_{i+1} \Delta w_{i+1} + M_{i+1}^T \Delta x_{i+1} = R_{w,i+1}, i \in \mathcal{N} \quad (8e)$$

$$-\Delta p_{i-1} + M_i \Delta w_i + \bar{X}_i \Delta x_i +$$

$$G_i^T \Delta u_i + A^T \Delta p_i = \bar{R}_{x,i}, i \in \bar{\mathcal{N}}, \quad (8f)$$

$$-\Delta p_{N-1} + M_N \Delta w_N + \bar{X}_N \Delta x_N = \bar{R}_{x,N}. \quad (8g)$$

Here we have defined

$$U_i := \Sigma_{\eta,i}^{-1} + \Sigma_{\lambda,i}^{-1} + \Sigma_{\omega,i}^{-1} + \Sigma_{v,i}^{-1}, \quad i \in \mathcal{N},$$

$$W_{i+1} := \Sigma_{\rho,i+1}^{-1} + \Sigma_{\xi,i+1}^{-1} + \Sigma_{\gamma,i+1}^{-1}, \quad i \in \mathcal{N},$$

$$\bar{X}_i := C^T(\Sigma_{\rho,i}^{-1} + \Sigma_{v,i}^{-1})C + D^T(\Sigma_{\gamma,i}^{-1} + \Sigma_{\omega,i}^{-1})D, \quad i \in \mathcal{N},$$

$$\bar{X}_N := C^T(\Sigma_{\rho,N}^{-1} + \Sigma_{v,N}^{-1})C,$$

$$G_i := -(\Sigma_{\omega,i}^{-1} + \Sigma_{v,i}^{-1})D, \quad i \in \mathcal{N},$$

$$M_{i+1} := C^T(\Sigma_{\rho,i+1}^{-1} - \Sigma_{\gamma,i+1}^{-1}), \quad i \in \mathcal{N},$$

and

$$\begin{aligned}
R_{u,i} &:= r_{u,i} + \Sigma_{\lambda,i}^{-1} r_{\lambda,i} + \\
&\quad \Sigma_{\omega,i}^{-1} r_{\omega,i} - \Sigma_{\eta,i}^{-1} r_{\eta,i} - \Sigma_{v,i}^{-1} r_{v,i}, \quad i \in \mathcal{N}, \\
R_{p,i} &:= -r_{p,i}, \quad i \in \mathcal{N}, \\
R_{w,i+1} &:= r_{w,i} + \Sigma_{\rho,i}^{-1} r_{\rho,i} + \\
&\quad \Sigma_{\xi,i+1}^{-1} r_{\xi,i+1} + \Sigma_{\gamma,i+1}^{-1} r_{\gamma,i+1}, \quad i \in \mathcal{N}, \\
\bar{R}_{x,i} &:= r_{x,i} + C^T (\Sigma_{\rho,i}^{-1} r_{\rho,i} - \Sigma_{\gamma,i}^{-1} r_{\gamma,i}) + \\
&\quad D^T (\Sigma_{v,i}^{-1} r_{v,i} - \Sigma_{\omega,i}^{-1} r_{\omega,i}), \quad i \in \bar{\mathcal{N}}, \\
\bar{R}_{x,N} &:= r_{x,N} + C^T (\Sigma_{\rho,N}^{-1} r_{\rho,N} - \Sigma_{\gamma,N}^{-1} r_{\gamma,N}).
\end{aligned}$$

Solving (8e) for Δw gives

$$\Delta w_{i+1} = W_{i+1}^{-1} (R_{w,i+1} - M_{i+1}^T \Delta x_{i+1}), \quad i \in \mathcal{N}. \quad (9)$$

Substituting back into (8) results in the equations

$$\begin{aligned}
B^T \Delta p_0 + U_0 \Delta u_0 &= R_{u,0}, \\
B^T \Delta p_i + U_i \Delta u_i + G_i \Delta x_i &= R_{u,i}, \quad i \in \mathcal{N}, \\
-\Delta x_1 + B \Delta u_0 &= R_{p,0}, \\
-\Delta x_{i+1} + A \Delta x_i + B \Delta u_i &= R_{p,i}, \quad i \in \mathcal{N}, \\
-\Delta p_{i-1} + X_i \Delta x_i + G_i^T \Delta u_i + A^T \Delta p_i &= R_{x,i}, \quad i \in \mathcal{N}, \\
-\Delta p_{N-1} + X_N \Delta x_N &= R_{x,N},
\end{aligned}$$

where

$$\begin{aligned}
X_{i+1} &:= \bar{X}_{i+1} - M_{i+1} W_{i+1}^{-1} M_{i+1}^T, \quad i \in \mathcal{N}, \\
R_{x,i+1} &:= \bar{R}_{x,i+1} - M_{i+1} W_{i+1}^{-1} R_{w,i+1}, \quad i \in \mathcal{N}.
\end{aligned}$$

As described in (Rao et al., 1998; Jørgensen et al., 2004; Wang and Boyd, 2010; Czyzyk et al., 1999), the equations above can be solved efficiently by a Riccati iteration procedure. With this approach the overall complexity of Algorithm 1 becomes of order $O(N(n_u^3 + n_y^3 + n_x^3))$ per iteration. In comparison, solving the system (6) by a general purpose method has order of complexity $O(N^3(n_u + n_y + n_x)^3)$. Thus, the computational cost per iteration is reduced by two orders of magnitude in N . Furthermore, a considerable improvement for systems of growing dimensions (n_u, n_x, n_y) is obtained as well. Therefore, a Riccati-based IPM is expected to run significantly faster than a conventional method and scale in a favourable way for both increasing system size and particularly for an increasing prediction horizon.

4.2 Homogeneous and Self-Dual Model

A drawback related to infeasible primal-dual IPMs such as Algorithm 1, is that the number of iterations can be very sensitive to the choice of initial point (Andersen et al., 1996). For MPC applications this presents a problem since reliability of the online solver is critical. A possible way to overcome the issue is to employ the homogeneous and self-dual model described in (Andersen et al., 1996; Xu et al., 1996; Ye et al., 1994; Jansen et al., 1993). Other than making it simple to find a suitable initial point, this approach facilitates easy detection of infeasibility, as well as strategies for warm starting (Skajaa et al., 2012). In the following, we show that the Riccati iteration procedure described above can be used for IPMs based on the homogeneous and self-dual model as well. For this class of algorithms, the optimality conditions are

$$V_H(t, p, z, s, \kappa, \tau) := \begin{bmatrix} F^T p + H^T z + g\tau \\ b\tau - Ft \\ c\tau - Ht - s \\ g^T t + b^T p + c^T z - \kappa \\ ZS\mathbf{1}_{m_I} \\ \tau\kappa \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

subject to $(z, s) \geq 0$ and $(\tau, \kappa) \in \mathbb{R}_+^2$. Consequently, the linear systems solved in a homogeneous and self-dual IPM corresponding to (6), can be written as

$$\begin{bmatrix} 0 & F^T & H^T & 0 & g & 0 \\ -F & 0 & 0 & 0 & b & 0 \\ -H & 0 & 0 & -I & c & 0 \\ g^T & b^T & c^T & 0 & 0 & -1 \\ 0 & 0 & S & Z & 0 & 0 \\ 0 & 0 & 0 & 0 & \kappa & \tau \end{bmatrix} \begin{bmatrix} \Delta t \\ \Delta p \\ \Delta z \\ \Delta s \\ \Delta \tau \\ \Delta \kappa \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ r_I \\ r_G \\ r_C \\ r_H \end{bmatrix}. \quad (10)$$

It can be proven that if $V_H(t^*, p^*, z^*, s^*, \tau^*, \kappa^*) = 0$, $(z^*, s^*) \geq 0$ and $(\tau^*, \kappa^*) \in \mathbb{R}_+^2$, then one of the following statements is true (Andersen et al., 1996):

- **Scaled solution optimal for (2)**
 $\tau^* > 0$ and $\kappa^* = 0 \Rightarrow V(t^*/\tau^*, p^*/\tau^*, z^*/\tau^*, s^*/\tau^*) = 0$.
- **Solution is certificate for infeasibility of (2)**
 $\tau^* = 0$ and $\kappa^* > 0 \Rightarrow$ either $-b^T p^* - c^T z^* > 0$ (implies primal infeasibility), or $g^T t^* < 0$ (implies dual infeasibility).

To solve (10) efficiently, we first decompose the system as in (Andersen et al., 2003). For this purpose it is convenient to write the equations as

$$F^T \Delta p + H^T \Delta z + g \Delta \tau = r_D, \quad (11a)$$

$$b \Delta \tau - F \Delta t = r_E, \quad (11b)$$

$$c \Delta \tau - H \Delta t - \Delta s = r_I, \quad (11c)$$

$$g^T \Delta x + b^T \Delta p + c^T \Delta z - \Delta \kappa = r_G, \quad (11d)$$

$$Z \Delta s + S \Delta z = r_C, \quad (11e)$$

$$\kappa \Delta \tau + \tau \Delta \kappa = r_H. \quad (11f)$$

We now solve (11d) for $\Delta \kappa$ and (11e) for Δs . This gives

$$\Delta \kappa = g^T \Delta x + b^T \Delta p + c^T \Delta z - r_G, \quad (12a)$$

$$\Delta s = Z^{-1} (r_C - S \Delta z), \quad (12b)$$

and the conditions

$$F^T \Delta p + H^T \Delta z + g \Delta \tau = r_D, \quad (13a)$$

$$b \Delta \tau - F \Delta t = r_E, \quad (13b)$$

$$c \Delta \tau - H \Delta t + Z^{-1} S \Delta z = \hat{r}_I, \quad (13c)$$

$$\kappa \Delta \tau + \tau g^T \Delta x + \tau b^T \Delta p + \tau c^T \Delta z = \hat{r}_H, \quad (13d)$$

in which we have defined

$$\hat{r}_I := r_I + Z^{-1} r_C, \quad \hat{r}_H := r_H + \tau r_G.$$

From (13d) it follows that

$$\Delta \tau = \frac{(\hat{r}_H - \tau(g^T \Delta x + \tau b^T \Delta p + \tau c^T \Delta z))}{\kappa}. \quad (14)$$

Based on (13a-13c) we can construct the linear systems

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & Z^{-1} S \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} r_D \\ r_E \\ \hat{r}_I \end{bmatrix}, \quad (15)$$

and

$$\begin{bmatrix} 0 & F^T & H^T \\ -F & 0 & 0 \\ -H & 0 & Z^{-1}S \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} = \begin{bmatrix} -g \\ -b \\ -c \end{bmatrix}, \quad (16)$$

such that a set of auxiliary variables f and h satisfy

$$\begin{bmatrix} \Delta t \\ \Delta p \\ \Delta z \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \Delta \tau, \quad (17)$$

which implies

$$\begin{bmatrix} g^T \Delta t \\ b^T \Delta p \\ c^T \Delta z \end{bmatrix} = \begin{bmatrix} g^T f_1 \\ b^T f_2 \\ c^T f_3 \end{bmatrix} + \begin{bmatrix} g^T h_1 \\ b^T h_2 \\ c^T h_3 \end{bmatrix} \Delta \tau.$$

Substituting into (14) yields

$$\Delta \tau = \frac{\hat{r}_H - \tau(g^T f_1 + b^T f_2 + c^T f_3)}{\kappa + \tau(g^T h_1 + b^T h_2 + c^T h_3)}.$$

Having determined f , g and $\Delta \tau$, the remaining variables can be recovered from (12) and (17). The major operations involved in solving (10) are therefore reduced to computing f and g from (15-16). This can be done efficiently by the Riccati iteration procedure described in Section 4.1. To determine both the affine step and the predictor-corrector step, 4 linear systems have to be solved in each iteration of our homogeneous and self-dual variant of Algorithm 1. In comparison, conventional IPMs only require solving 2 linear systems per iteration. Since the system matrix is constant in each iteration however, all the major computations are only involved in solving the first system. We therefore expect the additional cost per iteration in the homogeneous and self-dual algorithm to be insignificant compared to the overall reduction in the number of iterations. To detect if a solution is optimal or infeasible, we use the following measures (Andersen et al., 2003)

$$\begin{aligned} \rho_E &:= \frac{\|b\tau - Ft\|_\infty}{\max(1, \| [F \ b] \|_\infty)}, \\ \rho_I &:= \frac{\|c\tau - Ht - s\|_\infty}{\max(1, \| [H \ I \ c] \|_\infty)}, \\ \rho_D &:= \frac{\|g\tau + F^T p + H^T z\|_\infty}{\max(1, \| [H^T \ F^T \ g] \|_\infty)}, \\ \rho_G &:= \frac{|g^T t - (-b^T p - c^T z) - \kappa|}{\max(1, \| [g^T \ b^T \ c^T \ 1] \|_\infty)}, \\ \rho_O &:= \frac{|g^T t - (-b^T p - c^T z)|}{\tau + | -b^T p - c^T z |}. \end{aligned}$$

An iterate $(t^k, p^k, z^k, s^k, \tau^k, \kappa^k)$ is classified as optimal if

$$\rho_E^k \leq \epsilon_E, \quad \rho_I^k \leq \epsilon_I, \quad \rho_D^k \leq \epsilon_D, \quad \rho_O^k \leq \epsilon_O,$$

and infeasible if $\tau^k \leq \epsilon_\tau \max(1, \kappa^k)$ and

$$\rho_E^k \leq \epsilon_E, \quad \rho_I^k \leq \epsilon_I, \quad \rho_D^k \leq \epsilon_D, \quad \rho_G^k \leq \epsilon_G,$$

Again ϵ_τ , ϵ_E , ϵ_I , ϵ_D , ϵ_O and ϵ_G are small user-defined tolerances.

4.3 Special Operators

To avoid forming F and G explicitly, operations involving these matrices are implemented as special operations. For $N = 2$ the optimization variables may be written as

$$\begin{aligned} t &= [u_0^T \ x_1^T \ w_1^T \ u_1^T \ x_2^T \ w_2^T]^T \\ p &= [p_1^T \ p_2^T]^T \\ z &= [\eta_0^T \ \eta_1^T \ \lambda_0^T \ \lambda_1^T \ v_0^T \ v_1^T \\ &\quad \omega_0^T \ \omega_1^T \ \gamma_1^T \ \gamma_2^T \ \rho_1^T \ \rho_2^T \ \xi_1^T \ \xi_2^T]^T. \end{aligned}$$

In this case, the special operations are

$$\begin{aligned} F^T p &= [p_1^T B \ p_2^T A - p_1^T 0 \ p_2^T B \ -p_2^T 0]^T, \\ H^T z &= \begin{bmatrix} \eta_0 - \lambda_0 + v_0 - \omega_0 \\ D^T(\omega_1 - v_1) + C^T(\gamma_1 - \rho_1) \\ -\gamma_1 - \rho_1 - \eta_1 \\ \eta_1 - \lambda_1 + v_1 - \omega_1 \\ C^T(\gamma_2 - \rho_2) \\ -\gamma_2 - \rho_2 - \eta_2 \end{bmatrix}, \end{aligned}$$

and

$$\begin{aligned} Ht &= [u^T \ -u^T \ u_0^T \ (u_1^T - Dx_1)^T \ -u_0^T \ (Dx_1 - u_1)^T \\ &\quad (Cx_1 - w_1)^T \ (Cx_2 - w_2)^T \ (-Cx_1 - w_1)^T \\ &\quad (-Cx_2 - w_2)^T \ -w^T]^T, \\ Ft &= [(Bu_0 - x_1)^T \ (Ax_1 + Bu_1 - x_2)^T]^T. \end{aligned}$$

Multiplications involving F and G can thus be implemented very cheaply.

5. RESULTS

In this section, economic MPC is applied to a simple power system. Moreover, the algorithm developed in this paper is compared to state of the art general purpose solvers. For this purpose, we introduce a mass-spring system which has been previously used in e.g. (Shahzaad et al., 2010; Wang and Boyd, 2010) for evaluating performance of IPMs.

5.1 Case Study - Energy System

To illustrate economic MPC of energy systems, we consider a collection of power plants in the form

$$Y_i(s) = \frac{1}{(\tau_i s + 1)^3} U_i(s), \quad i = 1, 2, \dots, M,$$

where $U_i(s)$ is the units of fuel supplied to power plant i , and $Y_i(s)$ is power produced by power plant i . This third order model has been validated against measurement data in (Edlund et al., 2010). The total power production is given by $Y_T(s) := \sum_{i=1}^M Y_i(s)$. In the following simulation, 4 units with different operational features are controlled. We have listed the controller parameters in Table 1. The objective is to keep the total power production within a certain range, while minimizing input costs. The output limits are time varying and therefore not included in the table. To formulate a controller based on (1), the system is realized in a discrete state space form. We use a sampling time of $T_s = 5$ seconds, which is adequate to capture the system dynamics. The optimal open-loop solution for a $N = 120$ time step prediction horizon is depicted in Fig. 1. As desired, the total power generation tracks a predefined interval. The cheapest plant accounts for a majority of the load, whereas more expensive and flexible units are used only whenever faster dynamics are required.

Table 1. The generators have limited input and input-rate. Fuel costs increase inversely with the time constants (flexibility is expensive).

	Plant #1	Plant #2	Plant #3	Plant #4
τ	60	40	20	10
p_k	1	2.5	5	10
q_k	100	100	100	100
\underline{u}_k	0	0	0	0
\bar{u}_k	15	10	5	2.5
$\Delta \underline{u}_k$	-7.5	-5	-2.5	-1.25
$\Delta \bar{u}_k$	7.5	5	2.5	1.25

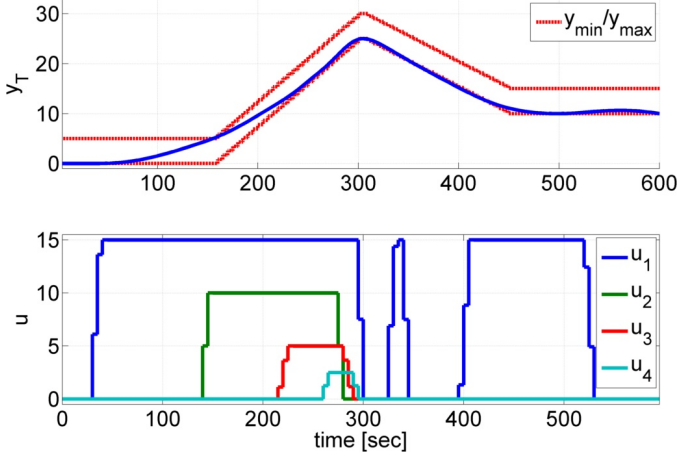


Fig. 1. Open-loop simulation of economic MPC applied to a simple power system.

Table 2. Algorithmic parameter settings for large-scale benchmark.

η	ϵ_E	ϵ_I	ϵ_D	ϵ_G	ϵ_O	ϵ_τ
0.995	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-8}	10^{-10}

5.2 Benchmark - Mass-Spring System

To evaluate the performance of our numerical algorithm, we set up and solve a number of growing economic MPC problems. The study is performed on an Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz with 4 GB RAM running a 64-bit Ubuntu 12.04.1 LTS operating system. For comparison, we use the following solvers:

- **SeDuMi**: Software package written in MATLAB and C for solving optimization problems over symmetric cones. The default solver is an implementation of a homogeneous and self-dual IPM (Sturm, 1999).
- **LIPSOL**: IPM for large-scale linear programming available through MATLAB's `linprog` interface (Zhang, 1995).
- **LPRsol**: Riccati-based MATLAB implementation of Algorithm 1 (the Riccati iteration procedure is implemented in a separate MEX-file).
- **LPRHsol**: Riccati-based MATLAB implementation of the homogeneous and self-dual variant of Algorithm 1 (the Riccati iteration procedure is implemented in a separate MEX-file).

Our parameter setting for **LPRsol** and **LPRHsol** is listed in Table 2. The initial point is $(t^0, p^0, s^0, z^0, \tau^0, \kappa^0) =$

Table 3. Economic MPC parameters for mass-spring system.

p_k	q_k	\underline{y}_k	\bar{y}_k	\underline{u}_k	\bar{u}_k	$\Delta \underline{u}_k$	$\Delta \bar{u}_k$
$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$	$\begin{bmatrix} 100 \\ 100 \\ 100 \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{20} \\ -\frac{1}{20} \\ -\frac{1}{20} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{20} \\ \frac{1}{20} \\ \frac{1}{20} \end{bmatrix}$	$\begin{bmatrix} 0 \\ 0 \end{bmatrix}$	$\begin{bmatrix} \frac{1}{2} \\ \frac{1}{2} \end{bmatrix}$	$\begin{bmatrix} -\frac{1}{8} \\ -\frac{1}{8} \end{bmatrix}$	$\begin{bmatrix} \frac{1}{8} \\ \frac{1}{8} \end{bmatrix}$

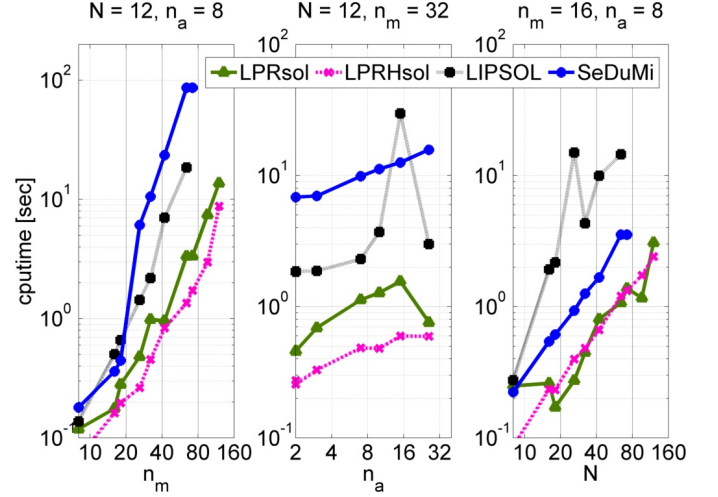


Fig. 2. CPU timings for economic MPC of mass-spring system.

$(0_n, 0_{m_E}, 1_{m_I}, 1_{m_I}, 1, 1)$. For **SeDuMi** and **LIPSOL** the default tolerance level and parameter setting is used. It has been verified that this approximately gives the same accuracy in the solution. The system used for benchmarks is a mass-spring system (Shahzad et al., 2010; Wang and Boyd, 2010). The system consists of n_m 1 kg masses connected by springs, and walls at the end. No damping is assumed and the spring constant is 1 N/m. Manipulable actuators are attached to each of the first $n_a \leq n_m$ masses. The objective of our controller is to keep the individual mass displacement within certain bounds, at a minimum cost. Initially, the mass displacement for all masses are 1 m. Costs are imposed for using the actuators and violating the displacement bounds. The actuators have a limited force and a limited rate of change. We have summarized the controller setup for $n_m = 3$ and $n_a = 2$ in Table 3. The parameters are held constant for all time steps. In Fig. 2, we have depicted computational results for problems of growing dimension. The figure shows that the structure-exploiting methods are about an order of magnitude faster than **SeDuMi** and **LIPSOL**. This difference increases as the problem size grows. Also notice that our homogeneous and self-dual IPM **LPRHsol**, outperforms the conventional IPM **LPRsol**. As illustrated in Fig. 3, this is due to its ability to consistently maintain a relatively low number of iterations. The same iteration pattern applies for **SeDuMi**, which is also based on the homogeneous and self-dual model.

6. CONCLUSIONS

In this paper, we have developed a structure-exploiting IPM, which combines a homogeneous and self-dual model with a highly specialized Riccati iteration procedure. Com-

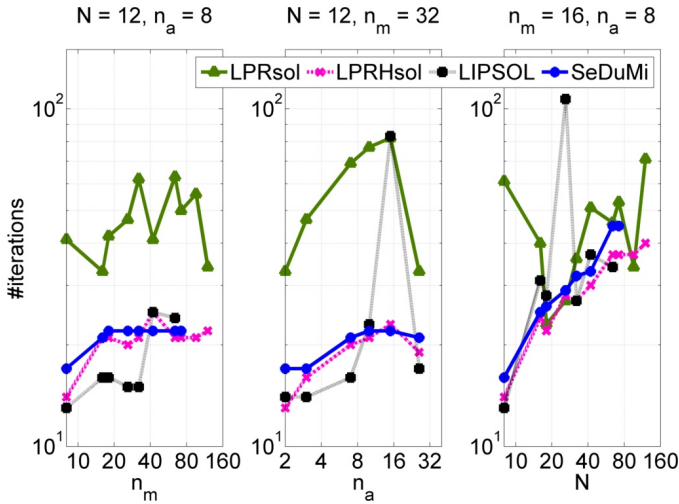


Fig. 3. Iteration counts for economic MPC of mass-spring system.

pared to general purpose solvers, the complexity of our algorithm is reduced by two orders of magnitude in the prediction horizon N , and a considerable improvement for growing system dimensions is obtained as well. Furthermore, the algorithm facilitates warm-starting and have a less fluctuating iteration pattern compared to conventional IPMs. Altogether, our solver is therefore well suited to handle real-time optimization of the structured linear programs which arise in economic MPC applications.

REFERENCES

- Alessio, A. and Bemporad, A. (2009). A Survey on Explicit Model Predictive Control. In L. Magni, D. Raimondo, and F. Allgöwer (eds.), *Nonlinear Model Predictive Control*, volume 384 of *Lecture Notes in Control and Information Sciences*, 345–369. Springer Berlin / Heidelberg.
- Andersen, E.D., Gondzio, J., Mészáros, C., and Xu, X. (1996). Implementation of Interior Point Methods for Large Scale Linear Programming. Papers 96.3, Ecole des Hautes Etudes Commerciales, Université de Genève.
- Andersen, E.D., Roos, C., and Terlaky, T. (2003). On implementing a primal-dual interior-point method for conic quadratic optimization. *Mathematical Programming*, 95(2), 249–277.
- Bemporad, A. and Patrino, P. (2012). Simple and Certifiable Quadratic Programming Algorithms for Embedded Linear Model Predictive Control. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 14–20. Noordwijkerhout, The Netherlands.
- Czyzyk, J., Mehrotra, S., Wagner, M., and Wright, S.J. (1999). PCx: an interior-point code for linear programming. *Optimization Methods and Software*, 11(1-4), 397–430.
- Edlund, K., Sokoler, L.E., and Jørgensen, J.B. (2009). A primal-dual interior-point linear programming algorithm for MPC. *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 351–356.
- Edlund, K., Mølbak, T., and Bendtsen, J.D. (2010). Simple Models for Model-based Portfolio Load Balancing Controller Synthesis. *IFAC Workshop Series*, 1.
- Ferreau, H.J., Bock, H.G., and Diehl, M. (2008). An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8), 816–830.
- Ferreau, H.J., Kozma, A., and Diehl, M. (2012). A Parallel Active-Set Strategy to Solve Sparse Parametric Quadratic Programs arising in MPC. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 74–79. Noordwijkerhout, The Netherlands.
- Hans-Bernd, D. and Ebenbauer, C. (2012). On a Class of Smooth Optimization Algorithms with Applications in Control. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 291–298. Noordwijkerhout, The Netherlands.
- Hartley, E.N., Jerez, J.L., Suardi, A., Maciejowski, J.M., Kerrigan, E.C., and Constantinides, G.A. (2012). Predictive Control of a Boeing 747 Aircraft using an FPGA. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 80–85. Noordwijkerhout, The Netherlands.
- Jansen, B., Roos, C., and Terlaky, T. (1993). *The Theory of Linear Programming: Skew Symmetric Self-dual Problems and the Central Path*. Reports of the Faculty of Mathematics and Informatics, Delft University of Technology. Fac., Univ.
- Jones, C.N., Domahidi, A., Morari, M., Richter, S., Ullmann, F., and Zeilinger, M. (2012). Fast Predictive Control : Real-time Computation and Certification. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 94–98. Noordwijkerhout, The Netherlands.
- Jørgensen, J.B., Frison, G., Gade-Nielsen, N.F., and Damman, B. (2012). Numerical Methods for Solution of the Extended Linear Quadratic Control Problem. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 187–193. Noordwijkerhout, The Netherlands.
- Jørgensen, J.B., Rawlings, J.B., and Jørgensen, S.B. (2004). Numerical Methods for Large Scale Moving Horizon Estimation and Control. In *Proceedings of Int. Symposium on Dynamics and Control Process Systems (DYCOPS)*, volume 7.
- Kerrigan, E.C., Jerez, J.L., Longo, S., and Constantinides, G.A. (2012). Number Representation in Predictive Control. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 60–67. Noordwijkerhout, The Netherlands.
- Kvasnica, M. and Fikar, M. (2010). Design and implementation of model predictive control using Multi-Parametric Toolbox and YALMIP. In *Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on*, 999–1004.
- Markus, K. and Findeisen, R. (2012). Parallel solution of model predictive control using the alternating direction multiplier method. In *Proc. IFAC Conf. Nonlinear Model Predictive Control (NMPC'12)*, 369–374. Noordwijkerhout, The Netherlands.
- Mattingley, J., Wang, Y., and Boyd, S. (2010). Code generation for receding horizon control. In *Computer-Aided Control System Design (CACSD), 2010 IEEE International Symposium on*, 985–992.
- Mehrotra, S. (1992). On the Implementation of a Primal-Dual Interior Point Method. *SIAM Journal on Optimization*, 2(4), 575–601.

- Nocedal, J. and Wright, S.J. (1999). *Numerical Optimization*. Springer Series in Operations Research. Springer.
- Pannocchia, G., Rawlings, J.B., and Wright, S.J. (2007). Fast, large-scale model predictive control by partial enumeration. *Automatica*, 43(5), 852–860.
- Potschka, A., Kirches, C., Bock, H.G., and Schlöder, J.P. (2010). Reliable solution of convex quadratic programs with parametric active set methods. Technical report, Interdisciplinary Center for Scientific Computing, Heidelberg University, Germany.
- Rao, C.V., Wright, S.J., and Rawlings, J.B. (1998). Application of Interior-Point Methods to Model Predictive Control. *Journal of Optimization Theory and Applications*, 99(3), 723–757.
- Shahzad, A., Kerrigan, E.C., and Constantinides, G.A. (2010). A fast well-conditioned interior point method for predictive control. *Decision and Control (CDC), 2010 49th IEEE Conference on*, 508–513.
- Skajaa, A., Andersen, E.D., and Ye, Y. (2012). Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems. *Mathematical Programming Computation*, 1–25.
- Sturm, J.F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, 11-12, 625–653.
- Summers, S., Jones, C.N., Lygeros, J., and Morari, M. (2011). A Multiresolution Approximation Method for Fast Explicit Model Predictive Control. *Automatic Control, IEEE Transactions on*, 56(11), 2530–2541.
- Vandenberghe, L., Boyd, S., and Nouralishadi, M. (2002). Robust Linear Programming and Optimal Control. In *15th IFAC World Congress*. Barcelona.
- Wang, Y. and Boyd, S. (2010). Fast Model Predictive Control Using Online Optimization. *Control Systems Technology, IEEE Transactions on*, 18(2), 267–278.
- Wright, S.J. (1987). *Primal-Dual Interior-Point Methods*. Miscellaneous Bks. Society for Industrial and Applied Mathematics.
- Xu, X., Hung, P.F., and Ye, Y. (1996). A simplified homogeneous and self-dual linear programming algorithm and its implementation. *Annals of Operations Research*, 62(1), 151–171.
- Ye, Y., Todd, M.J., and Mizuno, S. (1994). An $O(\sqrt{nL})$ -iteration Homogeneous and Self-dual Linear Programming Algorithm. *Mathematics of Operations Research*, 19(1), 53–67.
- Zhang, Y. (1995). Solving Large-Scale Linear Programs by Interior-Point Methods Under the MATLAB Environment. Technical report, Department of Mathematics and Statistics, University of Maryland, Baltimore County, Baltimore, MD.